In the claims:

1. (currently amended) A method for routing a data packet comprising:

receiving the packet at a first device;

producing ~~a plurality of threads~~ at least one thread associated with routing the packet, ~~each~~ the thread ~~being~~ including a sequence of instructions ~~that facilitates packet routing and that is independently executable with respect to other ones of the threads~~;

assigning a thread identifier (TID) to each of the threads and maintaining an activity status for each thread;

for each instruction ~~thread~~, selecting a pipeline from a plurality of processing pipelines such that no instruction in the selected pipeline depends on a preceding instruction because every instruction in the selected pipeline is associated with a different thread; and

executing the instructions in order to route the packet, including transmitting the packet from the first device to a second device~~, at least some of which are specialized, and forwarding that thread to the selected pipeline, such that processing of each packet is divided into multiple independent threads which are processed by multiple pipelines, and such that delay in processing of a first packet routing thread in a first pipeline does not affect processing of a second packet routing thread in a second pipeline~~.

2. (previously presented) The method according to claim 1, further comprising: transferring the first thread from an input buffer to a packet task manager; dispatching the first thread from the packet task manager to an analysis machine; classifying the first thread in the analysis machine; and modifying and forwarding the first thread in a packet manipulator.

3. (previously presented) The method according to claim 1, wherein the activity status indicates that a status of the associated thread is one of active, inactive or waiting.

4. (cancelled)

5. (currently amended) An apparatus for routing a packet comprising:

a memory for storing:

~~a plurality of threads~~ at least one thread associated with routing the packet, each thread ~~being~~ including a sequence of instructions ~~that facilitates packet routing and that is independently executable with respect to other ones of the threads~~;

a unique Thread Identifier (TID) for each thread; and

an activity status for each thread; and

an analysis machine including a plurality of pipelines, ~~at least some of which are specialized,~~ the analysis machine selecting a pipeline for each instruction such that no instruction in the selected pipeline depends on a preceding instruction because every instruction in the selected pipeline is associated with a different thread ~~thread~~ and forwarding that instruction ~~thread~~ to the selected pipeline ~~such that processing of each packet is divided into multiple independent threads which are processed by multiple pipelines, and such that delay in processing of a first packet routing thread in a first pipeline does not affect processing of a second packet routing thread in a second pipeline~~.

6. (previously presented) The apparatus according to claim 5, wherein one pipeline is dedicated to directly manipulating individual data bits of a bit field a packet task manager operationally connected to said analysis machine, and a packet manipulator operationally connected to said analysis machine.

7. (original) The apparatus according to claim 6, wherein said analysis machine is multi-threaded.

8. (original) The apparatus according to claim 6, wherein said analysis machine has 32 threads.

9. (original) The apparatus according to claim 6, further comprising: a packet task manager operationally connected to said analysis machine; a packet manipulator operationally connected to said analysis machine; and a global access bus including a master request bus and a slave request bus separated from each other and pipelined.

10. (original) The apparatus according to claim 6, further comprising: an external memory engine operationally connected to said analysis machine; and a hash engine operationally connected to said analysis machine.

11. (previously presented) The apparatus according to claim 9, further comprising: packet input global access bus program code , stored in a computer readable memory and operable when executed to control a flow of data packet information from a flexible input data buffer to the analysis machine.

12. (previously presented) The apparatus according to claim 9, further comprising: packet data global access bus program code, stored in a computer readable memory and operable when executed to control a flow of packet data between a flexible data input bus and the packet manipulator.

13. (previously presented) The apparatus according to claim 9, further comprising: statistics data global access bus software code used for connection of the analysis machine to the packet manipulator.

14. (previously presented) The apparatus according to claim 9, further comprising: private data global access bus software code used for connection of the analysis machine to an internal memory engine submodule.

15. (previously presented) The apparatus according to claim 9, further comprising: lookup global access bus software code used for connection of the analysis machine to an internal memory engine submodule.

16. (original) The apparatus according to claim 9, further comprising: results global access bus software code used for providing flexible access to an external memory.

17. (previously presented) The apparatus according to claim 5, wherein the activity status indicates that the associated multi-IP packet thread status is one of active, inactive or waiting.

18. (original) The apparatus according to claim 9, further comprising: a bi-directional access port operationally connected to said analysis machine; an input buffer operationally connected to said analysis machine; and an output buffer operationally connected to said analysis machine.

19. (new) The method of claim 1 wherein executing the instructions includes executing computational instructions including an arithmetic, boolean, or shift operation.

20. (new) The method of claim 1 wherein executing the instructions includes executing multi-argument instructions.

21. (new) The method of claim 1 wherein executing the instructions includes executing atomic instructions including a read-modify-write operation on data in private memory or statistics memory.

22. (new) The method of claim 1 wherein executing the instructions includes executing flow control instructions including a jump or branch operation.

23. (new) The method of claim 1 wherein executing the instructions includes executing load and store instructions to move data between memory spaces and registers.

24. (new) The method of claim 1 wherein executing the instructions includes executing search engine instructions including a filtering, lookup, or memory access operation.

25. (new) The apparatus of claim 5 wherein the selected pipeline executes computational instructions including an arithmetic, boolean, or shift operation.

26. (new) The apparatus of claim 5 wherein the selected pipeline executes multi-argument instructions.

27. (new) The apparatus of claim 5 wherein the selected pipeline executes atomic instructions including a read-modify-write operation on data in private memory or statistics memory.

28. (new) The apparatus of claim 5 wherein the selected pipeline executes flow control instructions including a jump or branch operation.

29. (new) The apparatus of claim 5 wherein the selected pipeline executes load and store instructions to move data between memory spaces and registers.

30. (new) The apparatus of claim 5 wherein the selected pipeline executes search engine instructions including a filtering, lookup, or memory access operation.